

OPEN ACCESS



RECEIVED
19 April 2020

REVISED
2 June 2020

ACCEPTED FOR PUBLICATION
16 June 2020

PUBLISHED
4 August 2020

Original content from
this work may be used
under the terms of the
[Creative Commons
Attribution 4.0 licence](#).

Any further distribution
of this work must
maintain attribution to
the author(s) and the
title of the work, journal
citation and DOI.



PAPER

Tree decompositions of real-world networks from simulated annealing

Konstantin Klemm

IFISC (CSIC-UIB), Palma de Mallorca, Spain

E-mail: klemm@ifisc.uib-csic.es

Keywords: network decomposition, exact algorithm, simulated annealing, tree decomposition, density of states

Supplementary material for this article is available [online](#)

Abstract

Decompositions of networks are useful not only for structural exploration. They also have implications and use in analysis and computational solution of processes (such as the Ising model, percolation, SIR model) running on a given network. Tree and branch decompositions considered here directly represent network structure as trees for recursive computation of network properties. Unlike coarse-graining approximations in terms of community structure or metapopulations, tree decompositions of sufficiently small width allow for exact results on equilibrium processes. Here we use simulated annealing to find tree decompositions of narrow width for a set of medium-size empirical networks. Rather than optimizing tree decompositions directly, we employ a search space constituted by so-called elimination orders being permutations on the network's node set. For each in a database of empirical networks with up to 1000 edges, we find a tree decomposition of low width.

1. Introduction

The analysis and modelling of complex systems involves complex *computational* tasks. In comparing empirical network structures with network models, for instance, one asks if they share common macroscopic behaviour under a percolation process or spin kinetics.

While theory of networks in complex systems has been enriched to include adaptive [1], temporally evolving [2, 3], and multi-layer [4, 5] structures, open questions remain also for processes on simple static structures. The Ising model in the macrocanonical ensemble, for instance, is fully captured by its density of states. Computing the exact density of states is not feasible in general. A subproblem of this task is to decide if there is a state at or above a given energy ϵ . Even this modest subproblem is equivalent to the NP-complete maximum cut problem on the given network [6, 7].

Computational problems on networks are made more feasible when effective size is reduced by coarse-graining, e.g. [8]. Several nodes with similar neighborhoods forming a community [9] or block [10] are lumped together into one representative node. Similar coarse-graining is applied in metapopulation models for epidemic spreading [11, 12] and opinion formation [13] where the nodes of the network are geographic locations rather than individuals. Coarse-graining amounts to lossy compression [14], however, so precision of results decreases with the amount of coarse-graining.

Here we advocate the study and development of exact methods [15, 16] rather than approximations and heuristics. In parallel to the development of exact methods, their applicability is to be explored by finding the networks and classes of networks that allow for efficient computation.

Tree networks allow for particularly simple recursive methods [17] where the network itself is used as the recursion tree. For non-tree networks, the same recursion equations are applied as a tree-like heuristic [18–21]. Recent research generalizes message passing to account for cycles to a certain extent. Radicchi and Castellano

[22] introduce corrections due to triangles only. Cantwell and Newman [23] consider expansion in cycle length leading to equations that can be solved exactly for some applications.

Rather different generalizations of the tree property, termed tree decompositions and branch decompositions are known in the areas of combinatorial optimization and graph algorithms [24–27]. A network has tree-width k if it can be split into smaller subnetworks with overlaps of at most k nodes, where this property holds again for the subnetworks; the tree representation of this split recursion is then called a tree decomposition of width k . See section 2 for the strict definition. Now consider a computation requiring time $\mathcal{O}(2^n)$ in terms of problem size n with a naive algorithm, e.g. exhaustively enumerating the 2^n spates of an n -spin Ising model. The computation drastically reduces to running time $\mathcal{O}(n2^k)$ when exploiting tree-width $k \ll n$. This makes exact computation feasible as long as there is a tree decomposition of sufficiently low width k .

This suggests a two-step process as a general *modus operandi* for solving a computational problem on a given network. (i) Find a tree decomposition of possibly low width. (ii) Solve the original problem with a method efficiently exploiting the low tree-width. Step (i) may fail, either because the given network does not have a tree decomposition of sufficiently low width or because we are unable to find it. Finding tree decompositions of optimally low width is in itself an NP-hard problem [24]. Thus we resort to a heuristic such as a greedy algorithm or stochastic optimization in step (i) [26, 28].

Here we demonstrate this two-step process for a collection of small and medium-size networks frequently employed as test structures. In step (i), we find tree decompositions of low width by simulated annealing [29]. Demonstrating step (ii), we use the tree decomposition and find the exact values of the networks' maximum Ising energy.

2. Mathematical background, methods, and data

2.1. Tree decompositions

Let $G = (V, E)$ be a graph with node set V and edge set E . Let \mathcal{B} a family of non-empty subsets of V . Each such $B \in \mathcal{B}$ is called a *bag*, being $B \subseteq V$. Furthermore, let T be a set of edges (unordered tuples) on \mathcal{B} , so that (\mathcal{B}, T) is a tree. Now this tree with node set \mathcal{B} and edge set T is called a *tree decomposition* of G if additionally the following two conditions hold.

- (a) For each edge $\{v, w\} \in E$, there is a bag $B \in \mathcal{B}$ so that $v \in B$ and $w \in B$.
- (b) The set of bags containing v induces a connected subgraph of (\mathcal{B}, T) .

In the literature (e.g. [26]), an additional requirement is that each node $v \in V$ is contained in at least one bag. This makes a difference only when there are isolated nodes; a non-isolated node must anyway appear in a bag due to condition (a). We stress also that a tree decomposition of the graph G is never a spanning tree of G . The graph and its tree decomposition have different node sets, V and \mathcal{B} .

Each graph $G = (V, E)$ trivially has a tree decomposition: simply take $\mathcal{B} = \{V\}$, a single bag containing all nodes, and $T = \emptyset$. Tree decompositions of use for recursive computation, however, need to achieve small bag size. The *width* w_∞ of a tree decomposition (\mathcal{B}, T) is the size of a largest bag, reduced by 1,

$$w_\infty(\mathcal{B}, T) = \max\{|B| : B \in \mathcal{B}\} - 1. \quad (1)$$

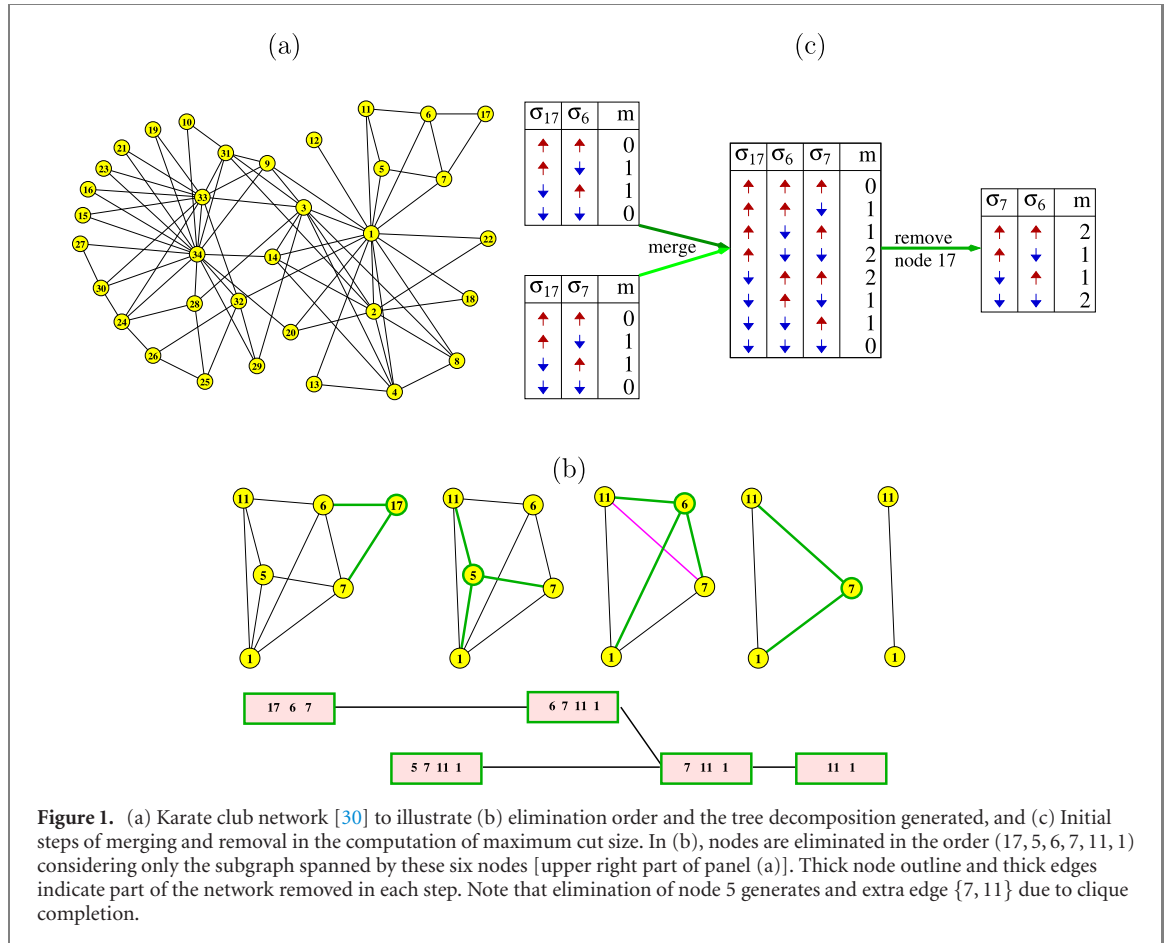
The tree-width of a graph G is the minimum of width over all tree decompositions of G . If G itself is a tree, G has a tree decomposition with a bag $B = \{v, w\}$ for each edge $\{v, w\} \in E$. Thus a tree has tree-width 1.

2.2. Elimination order

Finding a tree decomposition of low width for a given graph is a hard problem in itself. Stochastic search methods like annealing and genetic algorithms are useful for this task. The application of these methods is made easier by encoding the search space [31] rather than operating on the tree decompositions themselves. Elimination order is one such encoding of tree decomposition, with the theory well explored [26]. For the purpose of optimization, the encoding by elimination orders has the benefit of restricting the search space. For a graph of n nodes, there are infinitely many tree decompositions but only $n!$ elimination orderings including the optimal ones.

Let us define the operation of *elimination* Λ . For graph $G = (V, E)$ and node $v \in V$, elimination of v from G generates a graph $\Lambda(G, v)$ by (i) adding all missing edges between neighbours of v , thus turning the neighbourhood of v into a clique and (ii) removing v and all its incident edges.

Now let $\pi = \pi_1, \pi_2, \dots, \pi_n$ be an ordering (permutation) of the nodes of the graph $G = (V, E)$ with $n = |V|$ nodes. For each $v \in V$, there is exactly one index $i \in [n]$ with $\pi_i = v$. The graph sequence of G under



elimination order π is $G^{(0)}, G^{(1)}, G^{(2)}, \dots, G^{(n)}$ generated from initial element $G^{(0)} = G$ by recursion $G^{(i)} = \Lambda(G^{(i-1)}, \pi_i)$ for $i \in \{1, 2, \dots, n\}$. $G^{(n)}$ is the empty graph.

An elimination order π for a graph $G = (V, E)$ gives rise to a tree decomposition (\mathcal{B}, T) of G in the following way: the set of bags is $\mathcal{B} = \{B_i : i \in \{1, \dots, n\}\}$. Now for each i , bag B_i contains π_i and its neighbours in $G^{(i)}$. Note that π_i may have neighbours in $G^{(i)}$ it does not have in G itself. For indices $i < j$, the tree has an edge $\{i, j\} \in T$ if and only if $j = \min\{k > i : \pi_k \in B_i\}$. This means that, for each $i \neq n$, bag B_i is connected to exactly one bag with higher index, being the bag of the earliest (w.r.t. π) neighbour. Figure 1(b) illustrates elimination and generation of a tree decomposition from elimination order for a subgraph of the Karate club network (figure 1(a)).

Some model networks grow incrementally by attaching a new node to an existing clique of given size m [32–34]. In this case considering the nodes in reverse order of addition yields a *perfect elimination order* whose application does not involve edge addition. In each elimination step, the neighbourhood of the eliminated vertex is a clique already by construction. Graphs with a perfect elimination order are *chordal*, exhibiting further useful features [35].

2.3. Simulated annealing

We use a standard setting for simulated annealing [29] as a Metropolis Markov chain [36] with a slow decrease of temperature. We initialize the elimination order π with a permutation drawn from the uniform distribution of permutations on the node set. In order to generate a proposal π' from π , select an index $i \in \{1, \dots, n-1\}$ uniformly at random and swap elements at positions i and $i+1$:

$$\pi'_j = \begin{cases} \pi_{i+1} & \text{if } j = i \\ \pi_i & \text{if } j = i+1 \\ \pi_j & \text{otherwise} \end{cases} \quad (2)$$

The proposal is accepted with probability

$$\min \{ \exp[\beta(t)(r(\pi) - r(\pi'))], 1 \} \quad (3)$$

where the cost function r assigns each elimination order a real number, see section.

The inverse temperature β depends on time t as $\beta(t) = ct$ with cooling speed c as parameter. Time is measured in sweeps, so the clock advances by $\Delta t = 1/(n-1)$ in every step of proposal and acceptance.

2.4. Cost functions

A large part of work on tree decompositions focuses on minimization of tree width [26], which translates to stochastic optimization with a cost function $w_\infty(\pi)$ on elimination order π . We know of the following two problems that motivate the use of cost functions different from $w_\infty(\pi)$.

- (a) As a maximum over all bags, $w_\infty(\pi)$ only varies at moves that involve the largest bags of the tree decomposition. This renders the cost of a solution equal to that of most of its neighbouring solutions, thus lacking local gradient information in the search space.
- (b) While the time needed for a computational task on a network has an upper bound in terms of tree width, the time actually depends on *all* bag sizes s_1, s_2, \dots, s_n of the tree decomposition, typically as $\sum_{i=1}^n 2^{s_i}$ for spin systems.

Clautiaux *et al* [37] propose a solution to problem 1 above in terms of a modified cost function

$$m(\pi) = w_\infty(\pi) + n^{-2} \sum_{i=1}^n s_i^2. \quad (4)$$

The additional sum accounts for the sizes of all bags while the dominating term remains w_∞ .

We introduce the cost function

$$w_\eta(\pi) = \log_\eta [n^{-1} \sum_{i=1}^n \eta^{s_i-1}]. \quad (5)$$

with a parameter $\eta > 1$. This is motivated by both items 1 and 2 above. For spin systems with computation time proportional to 2^{s_i} for bag i , the parameter value $\eta = 2$ is to be chosen. Note that w_η has the tree-width w_∞ as a limit $\eta \rightarrow \infty$ if there is a unique maximum bag size.

2.5. Solving maximum cut using an elimination order

Given a network $G = (V, E)$, the maximum cut problem asks to find the largest bipartite subgraph of G . In other words, we are looking for a partition of the node set V into disjoint subsets V_1 and V_2 to maximize the number of edges running between V_1 and V_2 . Assigning set membership in V_1 or V_2 is equivalent to choosing a spin value $\sigma_v \in \{-1/2, +1/2\}$ for each node $v \in V$. Then a solution of the max-cut problem is a spin configuration $(\sigma_v)_{v \in V}$ that maximizes the number of edges between nodes with unequal spins,

$$H(\sigma) = \sum_{\{v,w\} \in E} |\sigma_v - \sigma_w|. \quad (6)$$

This amounts to finding the ground state of the spin glass with Hamiltonian H where each edge is an antiferromagnetic bond of unit weight. Equivalently we look for a maximum energy state of the Ising (all ferromagnetic) model on the same network. The maximum cut problem is NP-hard [7]. Therefore we do not know a general solver working in time polynomial in the size of the network.

For computing an exact solution, we first decompose the problem on the whole network into those on small feasible subproblems, namely one subproblem for each edge. For illustration, the leftmost boxes in figure 1(c) show the solutions on the subproblems for edge $\{6, 17\}$ and for edge $\{7, 17\}$ in the Karate club network. Solutions depend on spin values σ . A single edge $\{v, w\}$ contributes one unit ($m = 1$) to the maximum cut if $\sigma_v \neq \sigma_w$, otherwise it does not contribute ($m = 0$). The solution on the whole network is constructed by merging solutions of subproblems. In figure 1(c), the result of one merge step is the solution of the subnetwork formed by edges $\{6, 17\}$ and $\{7, 17\}$. It is now the only subproblem containing node 17, so subsequent merge operations will not depend on this node. By maximizing m over $\sigma_{17} = \uparrow$ and $\sigma_{17} = \downarrow$, the dependence on node 17 can be removed. This reduces the size of the partial solution from 8 to 4 rows, see figure 1(c). In general, merging subproblems and removal of node dependence according to an elimination order of width w_∞ ensures that partial solution sizes stay bounded by 2^{w_∞} .

Let us now describe the method more formally, using an elimination order π . During each phase of the computation, \mathcal{F} contains a set of functions storing partial results on cut sizes. Each function $f \in \mathcal{F}$ is based on a subset V_f of the node set. The domain of the function is the set $\{-1/2, +1/2\}^{V_f}$ of all spin configurations on V_f . Each such spin configuration is assigned the maximum cut size possible.

Initially, \mathcal{F} contains one function f for each edge $\{v, w\} \in E$ of the network. There are $2 \times 2 = 4$ spin configurations on the nodes v and w . If $\sigma_v \neq \sigma_w$, then $f(\sigma_v, \sigma_w) = 1$; otherwise $f(\sigma_v, \sigma_w) = 0$. Now the

Table 1. Networks considered and results obtained. Next to network name and reference, columns w_2 and w_∞ give the lowest values found by simulated annealing. For each network, minima of w_2 and w_∞ are taken from 10^6 sweeps of 80 independent runs, 10 runs for each of cooling speed value $c \in \{10^{-5}, 3 \times 10^{-4}, 10^{-4}, \dots, 3 \times 10^{-2}\}$. The rightmost column is the exact number b of edges in a maximum cut (edge-maximal bipartite subgraph) on the network. Networks have $|V|$ nodes and $|E|$ edges.

Network	Reference	w_2	w_∞	$ V $	$ E $	b
Karate club	[30]	3.0572	5	34	78	61
Social 3	[38]	5.3208	7	32	80	61
Protein 2	[38]	3.7212	6	53	123	92
Social 1	[38]	6.2729	10	67	142	111
Dolphins	[39]	6.4244	10	62	159	122
S 208	[40]	3.6738	6	122	189	168
<i>E. Coli</i> , transc.	[41]	4.3316	7	97	212	163
Protein 1	[38]	3.5963	5	95	213	160
Les Misérables	[42]	5.6124	9	77	254	169
College football	[43]	30.5677	34	115	316	
S 420	[40]	4.6018	9	252	399	354
Political books	[44]	9.4579	13	105	441	309
David Copperfield	[45]	23.8955	28	112	425	330
S 838	[40]	5.2136	9	512	819	724
Network science	[45]	3.3472	8	379	914	636

computation proceeds in a loop, running loop index i from 1 to $n = |V|$ and using the given elimination order $\pi = (\pi_1, \pi_2, \dots, \pi_n)$. In each round of the loop

- Find all functions $f \in \mathcal{F}$ with $\pi_i \in V_f$ and replace them by the merged function f' . See below for details of merging.
- Remove node π_i from the function f' generated at step 1. See below for details of node removal.

Note that merging in step 1 and removal in step 2 do not commute. Before removing a node r , all functions containing r need to be merged into a single one.

2.5.1. Merging functions

Two functions $f, g \in \mathcal{F}$ are merged into function h having the joint node set $V_h = V_f \cup V_g$. For each spin configuration $\sigma = (\sigma_v)_{v \in V_h}$,

$$h(\sigma) = f(\sigma) + g(\sigma) . \quad (7)$$

This describes the merging of two functions. For merging more than two functions, as happens in general at step 1 above, merging is applied several times. The operation is commutative, i.e. the result does not depend on the order of merging.

2.5.2. Removing nodes from functions

For a function $f \in \mathcal{F}$ and node $r \in V_f$, the removal of r from f generates a function $f^{\downarrow r}$ with $V_{f^{\downarrow r}} = V_f \setminus \{r\}$. For each spin configuration $\sigma = (\sigma_v)_{v \in V_{f^{\downarrow r}}}$, we have

$$f^{\downarrow r}(\sigma) = \max\{f(\sigma, \sigma_r = -1/2), f(\sigma, \sigma_r = +1/2)\} . \quad (8)$$

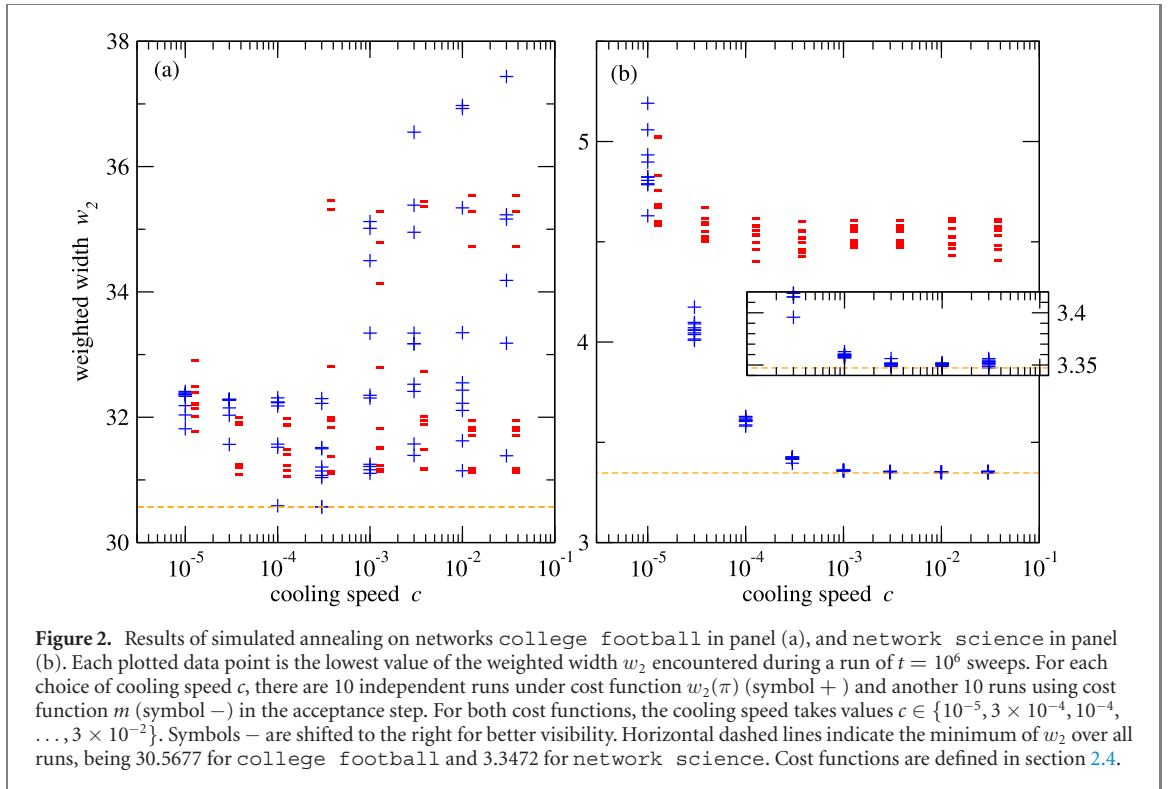
Independence from the variable σ_r is thus obtained by taking the maximum of the edge count with respect to the two spin orientations at node r .

2.6. Network data

We use the set of networks compiled by Radicchi and Castellano [22] restricting it to those 15 with less than 1000 edges. See table 1. These networks are frequently employed as examples and test cases in complex systems research.

3. Results

As a first step in exploring elimination orders by simulated annealing, we compare the effects of the choice of cost function, see also section 2.4. Figure 2 is a scatter plot of minimal w_2 values obtained in simulated



annealing runs. Better performance is obtained with optimization under cost function w_2 as compared to cost function m . The following optimization runs on all networks in the data set are thus performed with cost function w_2 only.

Table 1 is the overview of results for all 15 networks considered. We find tree decompositions with $w_2 < 10$ and $w_\infty \leq 13$, except for the networks *College football* with $w_2 \approx 30.6$ and *David Copperfield* ($w_2 \approx 23.9$).

The tree decompositions obtained by optimization enable us to calculate the exact maximum cut size for all but one of the networks. For the network *College football* with the tree decomposition of width 34, the computation runs out of memory on the machine currently used. The fraction of edges $b/|E|$ contained in a maximum cut is between 0.75 and 0.80 for most of the networks. Exceptions are the electronic circuits S 208, S 420, and S 838 with $b/|E| \approx 0.888$, confirming these networks to be close to bipartite [40].

4. Discussion

Originating in computer science and discrete mathematics, the concept of tree decompositions helps to exploit sparseness and tree-like structure in the analysis of network systems. The purpose of this contribution is an exposition of tree decompositions from the perspective of complex systems in physics. There are three results. (i) A simple method for finding tree decompositions by simulated annealing, tested on real networks; (ii) The resulting tree decompositions, encoded as node elimination orders, see the supplement (<https://stacks.iop.org/JPCOMPLEX/03/035003/mmedia>) [46]; and (iii) an example of an exact computation (maximum cut) made feasible by the use of tree decompositions.

Resulting from simulated annealing, the best tree decompositions we found are not necessarily optimal ones yet. In fact, the annealing method may be improved to become more suitable for the particular search space, e.g. with an adaptive cooling schedule [47]. Sometimes pictured as simulated annealing with ongoing restart, parallel tempering [48] may lead to better tree decompositions. Another ingredient for the method is to also compute a lower bound on tree-width [27]. Then the search for a good tree decomposition can be stopped once we are sufficiently close to the lower bound.

Tree decompositions enable us to efficiently analyze results for networks with respect to processes running on them. The method for the maximum-cut size and maximum Ising energy presented here is the basis for obtaining the exact partition function and complete density of states for Ising and Potts models [49]. For models of epidemic spreading and percolation [50], tree decompositions support the computation of cluster size distributions and epidemic thresholds as well [51, 52].

Acknowledgments

Funding from MINECO through the Ramón y Cajal program and through project SPASIMM, FIS2016-80067-P (AEI/FEDER, EU) is acknowledged. The data that support the findings of this study are openly available at the following https://ifisc.uib-csic.es/users/klemm/supplement_klemm_elorder.txt.

ORCID iDs

Konstantin Klemm  <https://orcid.org/0000-0002-0855-7474>

References

- [1] Gross T and Blasius B 2008 Adaptive codevolutionary networks: a review *J. R. Soc., Interface* **5** 259–71
- [2] Masuda N and Lambiotte R 2016 *A Guide to Temporal Networks* (Series on Complexity Science) (Singapore: World Scientific)
- [3] Masuda N, Klemm K and Eguíluz V M 2013 Temporal networks: slowing down diffusion by long lasting interactions *Phys. Rev. Lett.* **111** 188701
- [4] Bianconi G 2018 *Multilayer Networks: Structure and Function* (Oxford: Oxford University Press)
- [5] Diakonova M, Nicosia V, Latora V and San Miguel M 2016 Irreducibility of multilayer network dynamics: the case of the voter model *New J. Phys.* **18** 023010
- [6] Karp R M 1972 Reducibility among combinatorial problems *Complexity of Computer Computations: Proc. of a Symp. on the Complexity of Computer Computations* ed R E Miller J W Thatcher and J D Bohlinger (Boston, MA: Springer) pp 85–103
- [7] Garey M R and Johnson D S 1979 *Computers and Intractability: A Guide to the Theory of NP-Completeness* 1st edn (San Francisco, CA: Freeman)
- [8] Masuda N 2009 Immunization of networks with community structure *New J. Phys.* **11** 123018
- [9] Fortunato S 2010 Community detection in graphs *Phys. Rep.* **486** 75–174
- [10] Doreian P, Batagelj V and Ferligoj A (ed) 2019 *Advances in Network Clustering and Blockmodeling* (New York: Wiley)
- [11] Sattenspiel L and Dietz K 1995 A structured epidemic model incorporating geographic mobility among regions *Math. Biosci.* **128** 71–91
- [12] Riley S 2007 Large-scale spatial-transmission models of infectious disease *Science* **316** 1298–301
- [13] Fernández-Gracia J, Suchecki K, Ramasco J J, San Miguel M and Eguíluz V M 2014 Is the voter model a model for voters? *Phys. Rev. Lett.* **112** 158701
- [14] Rosvall M and Bergstrom C T 2007 An information-theoretic framework for resolving community structure in complex networks *Proc. Natl Acad. Sci.* **104** 7327–31
- [15] Lucet C and Manouvrier J-F 1999 Exact methods to compute network reliability *Statistical and Probabilistic Models in Reliability* ed D C Ionescu and N Limnios (Boston, MA: Birkhäuser) pp 279–94
- [16] Woeginger G J 2003 Exact algorithms for np-hard problems: a survey *Combinatorial Optimization—Eureka, You Shrink!: Papers Dedicated to Jack Edmonds 5th Int. Workshop* (Aussois, France, 5–9 March 2001) Revised Papers ed M Jünger and G Reinelt Rinaldi G (Berlin: Springer) pp 185–207
- [17] Pearl J 1982 Reverend bayes on inference engines: a distributed hierarchical approach *Proc. of the American Association of Artificial Intelligence National Conf. on AI* pp 133–6
- [18] Mézard M, Parisi G and Zecchina R 2002 Analytic and algorithmic solution of random satisfiability problems *Science* **297** 812–5
- [19] Karrer B, Newman M E J and Zdeborová L 2014 Percolation on sparse networks *Phys. Rev. Lett.* **113** 208702
- [20] Hamilton K E and Pryadko L P 2014 Tight lower bound for percolation threshold on an infinite graph *Phys. Rev. Lett.* **113** 208701
- [21] Moore S and Rogers T 2020 Predicting the speed of epidemics spreading in networks *Phys. Rev. Lett.* **124** 068301
- [22] Radicchi F and Castellano C 2016 Beyond the locally treelike approximation for percolation on real networks *Phys. Rev. E* **93** 030302
- [23] Cantwell G T and Newman M E J 2019 Message passing on networks with loops *Proc. Natl Acad. Sci.* **116** 23398–403
- [24] Arnborg S, Corneil D G and Proskurowski A 1987 Complexity of finding embeddings in a k-tree *SIAM J. Algebr. Discrete Methods* **8** 277–84
- [25] Robertson N and Seymour P D 1991 Graph minors. x. obstructions to tree-decomposition *J. Comb. Theory. B* **52** 153–90
- [26] Bodlaender H L and Koster A M C A 2010 Treewidth computations i. Upper bounds *Inf. Comput.* **208** 259–75
- [27] Bodlaender H L and Koster A M C A 2011 Treewidth computations ii. Lower bounds *Inf. Comput.* **209** 1103–19
- [28] Kjærulff U 1992 Optimal decomposition of probabilistic networks by simulated annealing *Stat. Comput.* **2** 7–17
- [29] Kirkpatrick S, Gelatt C D and Vecchi M P 1983 Optimization by simulated annealing *Science* **220** 671–80
- [30] Zachary W W 1977 An information flow model for conflict and fission in small groups *J. Anthropol. Res.* **33** 452–73
- [31] Klemm K, Mehta A and Stadler P F 2012 Landscape encodings enhance optimization *PloS One* **7** e34780
- [32] Dorogovtsev S N, Mendes J F F and Samukhin A N 2001 Size-dependent degree distribution of a scale-free growing network *Phys. Rev. E* **63** 062101
- [33] Klemm K and Eguíluz V M 2002 Highly clustered scale-free networks *Phys. Rev. E* **65** 036123
- [34] Eguíluz V M and Klemm K 2002 Epidemic threshold in structured scale-free networks *Phys. Rev. Lett.* **89** 108701
- [35] Golumbic M C 2004 *Algorithmic Graph Theory and Perfect Graphs* (Annals of Discrete Mathematics, Vol 57) (Amsterdam: North-Holland)
- [36] Toral R and Colet P 2014 *Stochastic Numerical Methods* (New York: Wiley)
- [37] Clautiaux F, Moukrim A, Nègre S and Carlier J 2004 Heuristic and metaheuristic methods for computing graph treewidth *Oper. Res.* **38** 13–26
- [38] Milo R, Itzkovitz S, Kashtan N, Levitt R, Shen-Orr S, Ayzenshtat I, Sheffer M and Alon U 2004 Superfamilies of evolved and designed networks *Science* **303** 1538–42
- [39] Lusseau D, Schneider K, Boisseau O J, Haase P, Slooten E and Dawson S M 2003 The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations—can geographic isolation explain this unique trait? *Behav. Ecol. Sociobiol.* **54** 396–405

- [40] Milo R, Shen-Orr S, Itzkovitz S, Kashtan N, Chklovskii D and Alon U 2002 Network motifs: simple building blocks of complex networks *Science* **298** 824–7
- [41] Mangan S and Alon U 2003 Structure and function of the feed-forward loop network motif *Proc. Natl Acad. Sci.* **100** 11980–5
- [42] Knuth D E 1993 *The Stanford GraphBase: A Platform for Combinatorial Computing* (New York: Association for Computing Machinery)
- [43] Girvan M and Newman M E J 2002 Community structure in social and biological networks *Proc. Natl Acad. Sci.* **99** 7821–6
- [44] Krebs V 2004 Unpublished, data posted online at <http://orgnet.com/>
- [45] Newman M E J 2006 Finding community structure in networks using the eigenvectors of matrices *Phys. Rev. E* **74** 036104
- [46] Supplementary information available online at https://ifisc.uib-csic.es/users/klemm/supplement_klemm_elorder.txt
- [47] Henderson D, Jacobson S H and Johnson A W 2003 The theory and practice of simulated annealing *Handbook of Metaheuristics* ed F Glover and G A Kochenberger (Berlin: Springer) pp 287–319
- [48] Earl D J and Deem M W 2005 Parallel tempering: theory, applications, and new perspectives *Phys. Chem. Chem. Phys.* **7** 3910–6
- [49] Klemm K 2020 Exact evaluation of ising partition function on networks (in preparation)
- [50] Grassberger P 1983 On the critical behavior of the general epidemic process and dynamical percolation *Math. Biosci.* **63** 157–72
- [51] Serra J P 2017 Exact computation of percolation cluster sizes in finite network *Master's Thesis* University of the Balearic Islands, Palma de Mallorca, Spain
- [52] Serra J P and Klemm K 2020 Exact computation of percolation cluster sizes (in preparation)